ActiveX control brings bit manipulation to Windows

Steve Hageman, Agilent Technologies, Santa Rosa, Ca

NOTHING COMPARES with the C language for working with bits. C provides a rich set of signed and unsigned number formats, along with many intrinsic bit-manipulation operators. However, most of the popular rapid-application-development Windows languages lack C's ability to easily work with bits. Visual Basic is such a language. Although it's hard to find a faster language to develop a small to midsized application in Windows, Visual Basic starts to show its weakness when it comes time to talk to hardware. Hardware programming is usually bit-oriented. That is, it's necessary to turn bits on and off or shift out serial streams to get the hardware to operate correctly. The ActiveX control serves just these types of bit-manipulation needs (**Figure 1**). The control includes functions for changing binary strings to numbers, a hex-output function, the ability to set and clear bits in a word, and the ever-needed shift-left and -right functions. As an example, many of the three-wire serial devices need to have a setup word shifted to them. Suppose you need to shift the setup word 0111 1101 first to an A/D converter to initiate a conversion on some channel. You can use the functions in the ActiveX control to easily effect the shift operation, as follows:

```
Setup_word = Bits ("01111101")
`Returns 125
For i = 0 to 7
    Val = ShiftRight_8(setup_word,0)
    `write val to the A/D here
next i
```

In the above example, val has the values 1, 0, 1, 1, 1, 1, 1, 0 during each iteration of the loop. The routine can then clock these bits to the A/D converter as required by the hardware. If the operation requires MSB first, you can use the ShiftLeft function. The SetBit and ClearBit functions are useful when using a port as clock and data lines, because you can set individual bits as needed instead of doing entire port writes. Any modern programming language that can use ActiveX controls, such as Agilent VEE, Visual Basic, Delphi, and others, can use the functions given here. You can download the ActiveX control from *EDN*'s Web site, www.ednmag.com. Click on "Search Databases" and then enter the Software Center to download the file for Design Idea #2534. The routine includes all the functions listed in **Figure 1**, plus a few more, with application examples. (DI #2534)

Figure 1

```
Function GetBit(ByVal x As Long, ByVal n As Integer) As Integer
Returns the value of bit n in input value x. Returns 1 or 0 if bit
is set or not. x = 1 to 16 bit, n = 0 = LSB.
Example: GetBit(16,5) returns 1.

Function Bits(ByVal inval As String) As Long
Given a  representation of a binary string, returns the
value. inval may be any length from 1 to 16 bits
Example: Bits("101") returns 5.

Function BitsStr(ByVal inval As Long, ByVal sizeof As Integer) As String
Given a number, returns with a representation of a binary string.
sizeof is the width of the return field (1 to 16 bits).
Example: BitsStr(82,8) returns "01010010"

Function HexStr(ByVal inval As Long, ByVal sizeof As Integer)
Given a number, returns with a representation of a hex string.
sizeof is the width of the return field (1 to 16 bits).
Example: HexStr(179,8) returns "B3"

Function ClearBit(ByVal x As Long, ByVal n As Integer) As Long
Clears bit position n in input x. Returns new x value.
x may be 1 to 16 bits, n = 0 = LSB
Example: ClearBit(16,4) returns 0.

Function SetBit(ByVal x As Long, ByVal n As Integer) As Long
Sets bit n in input value x. Returns new  x. x may be any
width 1 to 16 bits, n = 0 = LSB.
Example: SetBit(0,4) returns 16

Function ShiftRight_8(ByRef x As Integer, ByVal y As Integer) As Integer
Shifts the 8 bit value x right by 1 place. Bit shifted in is y.
Returns bit shifted out.
Example: ShiftRight_8(129,1) Returns 1 and the new value for x (was 129)
is 192.

Function ShiftRight_16(ByRef x As Long, ByVal y As Integer) As Integer
Shifts the 16 bit value x right by 1 place. Bit shifted in is y.
Returns bit shifted out.
Example: ShiftRight_16(1,1) Returns 1 and the new value for x (was 1)
is 32768.

Function ShiftLeft_8(ByRef x As Integer, ByVal y As Integer) As Integer
Shifts the 8 bit value x left by 1 place. Bit shifted in is y.
Returns bit shifted out.
Example: ShiftLeft_8(1,0) returns 0 and the new value for x (was 1)
is 2.

Function ShiftLeft_16(ByRef x As Long, ByVal y As Integer) As Integer
Shifts the 16 bit value x left by 1 place. Bit shifted in is y.
Returns bit shifted out.
Example: ShiftLeft_16(32768,1) returns 1 and the new value for x
(was 32768) is 1

Function RotateRight_8(ByVal x As Integer) As Integer
Rotates the 8 bit value x right by 1 place. Returns new value.
Example: RotateRight_8(1) returns 128.

Function RotateRight_16(ByVal x As Long) As Long
Rotates the 16 bit value x right by 1 place. Returns new value.
Example: RotateRight_16(1) returns 32768.

Function RotateLeft_8(ByVal x As Integer) As Integer
Rotates the 8 bit value x left by 1 place. Returns new value.
Example: RotateLeft_8(64) returns 128

Function RotateLeft_16(ByVal x As Long) As Long
Rotates the 16 bit value x left by 1 place. Returns new value.
Example: RotateLeft_16(32769) returns 3.

End.....
```

An ActiveX control offers many handy functions for bit manipulation.